2D Channel Identification, Equalization and Detection for Holographic Data Storage Systems

Sheida Nabavi

2005

Advisor: Prof. Kumar

2D Channel Identification, Equalization and Detection for Holographic Data Storage Systems

M.S. Project Report

by

Sheida Nabavi

Advisor: Prof. B.V.K. Vijaya Kumar

Carnegie Mellon University

Electrical and Computer Engineering Department

April 2005

Acknowledgements

Completion of this project report was certainly not possible without the help of all the valuable people around me. First I would like to thank professor Kumar for his patience and guidance throughout the course of this research effort. I would also wish to thank my colleagues in our research group, especially Lakshmi Ramamoorthy and Zongwang Li for their help and friendship.

I also would like to express my appreciation to Inphase Technologies for providing real data and generously addressing our technical issues.

Finally I would like to thank my husband Reza H. Sheikhi for his support and encouragement. My deepest gratitude goes to my parents who keep believing me and providing me with the best possible education.

Abstract

Holographic data storage is a promising data storage technology because of its potential for high data storage density (more than 1 Tbyte/in^2), high data rate (more than 1 Gb/s) and short access time (less than 1μs). In contrast to conventional data storage systems, holographic data storage systems use a parallel two-dimensional or page-like format in recording and retrieval.

In this project, we have investigated different methods of 2D channel identification, equalization and detection for holographic data storage channel. To evaluate different 2D equalization and detection methods, we needed the channel model. Therefore, we first developed a physical channel model based on the physical impairments. We evaluated the performance of this physical channel model using 60 real recorded and retrieved pages, provided by Inphase Technologies. The advantage of the physical channel model is that it allows us to control the impairment amounts and study their impact. We also investigated linear and nonlinear channel identification methods and compared their performance. We investigated the minimum mean square error (MMSE) equalizer, the zero forcing equalizer (ZFE), the adaptive decision feedback equalizer (ADFE) and the adaptive Volterra equalizer (AVE). For detection methods, we investigated fixed threshold, adaptive threshold, log likelihood ratio (LLR) and iterative detectors. To evaluate the performance of these equalization and detection techniques real pages as well as simulated pages (using physical channel model) have been used. The results show that MMSE equalizer outperforms the other equalizer for low signal to noise ratio (SNR) real data as well as simulated data. Iterative detector works well for higher SNR simulated page but adaptive threshold detectors performs better for real data.

Chapter 1

Introduction

In this chapter, first the basic principles and a brief history of holographic data storage system (HDS) are provided. Then, we describe the main components of HDS. Finally, we explain the motivation for investigating 2D signal processing for HDS.

1.1 What is holography?

The physical principles of holography involve the recording of the interference pattern, formed by interfering two coherent wavefronts (loosely called "beams") of light, and the subsequent illumination of that recorded pattern by one of the beams to recreate the other beam.

The intersection of two wavefronts creates an interference pattern of bright and dark regions. The interference pattern is stored as chemical and/or physical changes (e.g., absorption, refractive index, thickness) in a photosensitive medium. The hologram is the image of the interference pattern stored within the medium. Light from one wavefront shining on the hologram reconstructs the data pattern (Fig. 1-1).

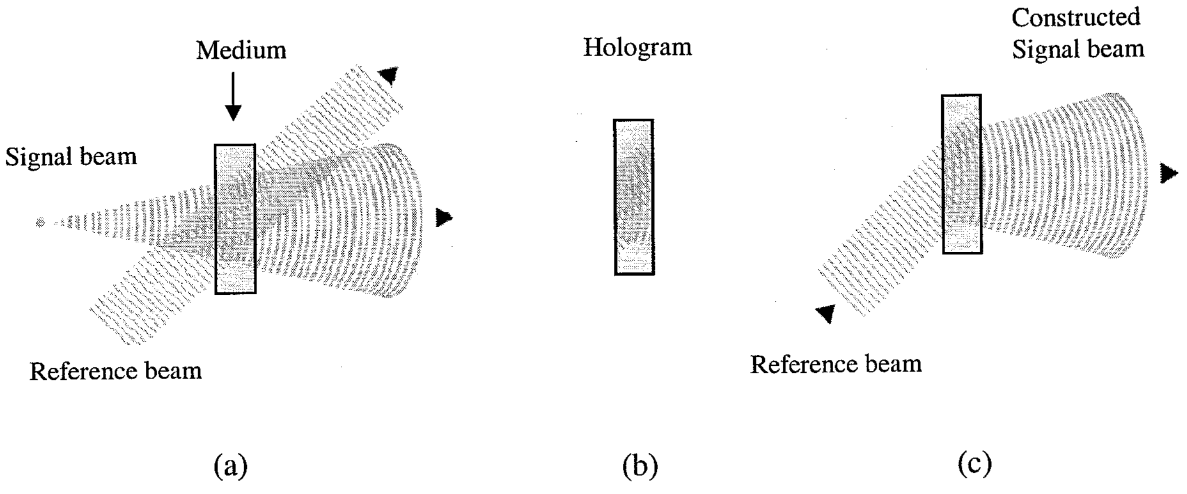


Figure1-1, (a) Recording, (b) Hologram, (c) Reading

In the specific case of holographic data storage, one wavefront is a signal wavefront containing a two dimensional pattern (or page) of 1s and 0s representing digital data. The other wavefront is a reference used to form the interference pattern and construct the signal wavefront. Both amplitude and phase information are stored and retrieved in HDS.

1.2 Brief history of holographic data storage system [1]

In 1891 Lippman demonstrated the first method of color photography by interfering a beam of light with its own reflection. The process was termed "interference heliography". While some amazing color photographs were made on black and white films, very little was done with this process until 1960s, when it was applied to white light holography using counter-propagating laser beams. Data storage based on Lippman process was explored briefly at IBM.

In 1948 Dennis Gabor made the insightful observation that the phase information in a wavefront could be recorded by interfering it with a coherent background. He showed that the original wavefront could be reproduced exactly by illuminating the recording with only the coherent background. The process was termed "holography".

It was not until the early 1960s with the invention of the laser that the technology became practical for storing and retrieving images. The concepts of holographic data storage (HDS) were established by Van Heerden . In his seminar paper in 1963 on the theory of optical information storage in solid materials, Van Heerden postulated that the recording of interference pattern in a three-dimensional medium could be used as a means of storing and retrieving information. The arrival of the laser provided the necessary coherent source. Although Van Heerden discussed the multiplexing of numerous holograms in a common volume by changing either the angle of the reference beam or the wavelength of both beams, it was not until 1973 that angular multiplexing was actually applied to the storage of information.

Although data storage using volume holography was proposed in 1970's, it has failed to become a commercial product mainly because of lack of suitable recording materials. Recently with significant improvement in supporting devices, optical systems and media materials and also using advanced signal processing, commercial HDS systems appear to be on the horizon.

1.3 Main components of HDS

The most important components in a HDS system are the spatial light modulator (SLM), optical lenses, the medium and the detector. The schematic of a HDS system is shown in Fig1-2.

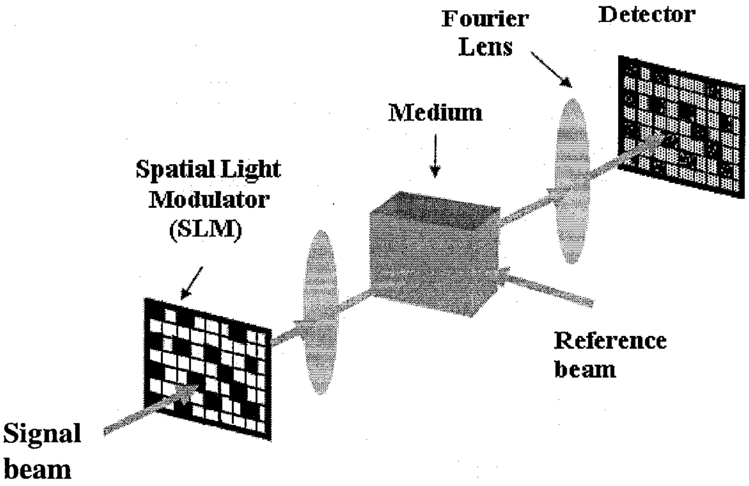


Figure1-2, Main component of HDS

Light from single laser beam is split into two beams, the signal beam (which carries the data) and the reference beam. During recording, a page of digital data (1's and 0's) is represented by the SLM as a pattern of bright and dark pixels. The exact number of data bits is determined by the pixel count of the SLM. Common size of SLM is 1024X1024 pixels. The signal beam passes through the SLM and becomes modulated in two dimensions by the pattern on the SLM. The SLM wave-front travels trough optical lenses and the interference of reference beam and the Fourier transform of the signal beam is stored in the medium. In order to read the data, the corresponding reference beam reconstructs the stored information. This wave-front passes through optical lenses so reconstructed information is back in spatial domain, and then is projected onto a detector array that reads the intensity of this wave-front.

The Fourier transform hologram is a useful configuration for holographic storage. In this arrangement the most useful components of the frequency spectrum can be formed into a relatively compact signal beam. This increases the information density and allows more effective use of the recording material. Another benefit is that Fourier transform holograms are less sensitive to misalignment and to imperfections in the SLM.

By using multiplexing, several data pages can be stored in the same volume of medium. Holograms read out by applying a reference beam identical to that used during recording. Most important multiplexing method are angular multiplexing where the angle of reference beam changes, wavelength multiplexing where the wavelength of reference beam changes and phase multiplexing where the of reference beam changes phase. With multiplexing methods, storing of 1000 data pages in the same volume of medium is possible.

1.4 Motivation for 2D signal processing for HDS

Due to the rapid growth of internet and digital communications, the need for storing huge amounts of data with fast access capabilities has increased dramatically. Conventional (i.e., magnetic and optical) data storage technologies are expected to reach their physical and engineering limits in the near future, and as a result investigating new, more unconventional data storage systems is necessary. Holographic data storage is a promising data storage technology because of its potential for high data storage density, high data rate and short access time. In contrast to conventional data storage systems, holographic data storage systems use a parallel two-dimensional or page-like format in recording and retrieval, leading to high data rates and short access times. The ability to multiplex several data pages into a given volume of the medium leads to potentially high volumetric storage densities.

Similar to conventional data storage systems, holographic data storage systems suffer from inter-symbol interference (ISI) and noise. Consequently advanced signal processing techniques need to be used to recover reliable data. Because of the two-dimensional (2D) nature of holographic data storage, 2D signal processing techniques have to be considered. 2D signal processing techniques are not limited to page-oriented data storage systems. With the increase in track density of conventional data storage systems (e.g., hard disk drives), the interference from adjacent tracks will increase leading to 2D ISI.

2D signal processing has an important role to recover reliable data for HDS. Utilizing advanced 2D signal processing techniques along with suitable supporting devices, optical systems and media materials are necessary to have a commercial HDS in near future.

1.5 Organization of this report

This report consists of six chapters. In first chapter, holographic data storage is introduced. The basic principles and brief history of HDS is described also. Chapter 2 consists of the explanation of the physical channel model. Chapter 3 covers channel identification. Linear minimum mean square error (LMMSE) estimation and nonlinear look-up-table are explained in this chapter followed by the comparison of these methods. In Chapter 4, the minimum mean square error (MMSE) equalizer, the zero forcing equalizer (ZFE), the adaptive decision feedback equalizer (ADFE) and the adaptive Volterra equalizer (AVE) are discussed. Their performances based on bit error rate (BER) are compared using real data and simulated data. Issues such as practicality and complexity are also considered in this chapter. Chapter 5 discusses the detection methods. Fixed threshold, adaptive threshold, log likelihood ratio (LLR) and iterative detection methods are analyzed. The BER results of these methods for raw input data are shown. The summary of this project is described in chapter 6.

Chapter 2

Physical Channel Model

In this chapter we describe a holographic data storage (HDS) physical channel model. This model is based on physical channel impairments and gives us a tunable simulator. Tunable simulator is important in investigating signal processing options. We discuss the channel impairments that we have considered in the channel model. Physical channel model is developed using MATLAB. A binary data page and channel parameters are the inputs and the detector (camera) output intensity page is the output of this simulator. All the parameters are tunable with the default values given by Inphase Technologies. The evaluation of this simulator is given in the next chapter.

2.1 Channel impairments and physical channel model

The channel impairments for main components (light source, SLM, storage medium and detector) of HDS are listed as following.

- **Light source**: Non-flat input illumination
- **SLM**: Finite contrast ratio, Non-full fill factor, Non-uniformity, Electronic noise and Phase mask
- **Storage medium**: Frequency plane aperture and Optical noise
- **Detector (Camera)**: Dark noise, Electronic noise, Non-full fill factor and Quantization

We model the impairments based on their physical functions and related mathematical equations. The details of these impairments and their models in the simulator are explained in this part. For physical channel model, a MATLAB-based simulator has been designed and implemented. This simulator starts with a page of binary (i.e., ones and zeros) data, applies all the above channel impairment and determines the camera output intensity. All the parameters are tunable. We have defined the default values for each of these parameters based on the information given by Inphase Technologies.

2.2 Light source

The input illumination is not flat. The coherent light source has a Gaussian wave-front. By using the central portion of the Gaussian, one can achieve a flatter illumination, but at the expense of light loss. Thus, there is a tradeoff between the light loss and input wave-front flatness. This non-flatness impairment is modeled as a two-dimensional centered Gaussian wave front. The width of this Gaussian wave front is determined based on the decrease of the intensity of the input illumination from the center to the corners of the SLM (Fig. 2-1). In the default model, 10% decrease in intensity, from the center to the corners is assumed.

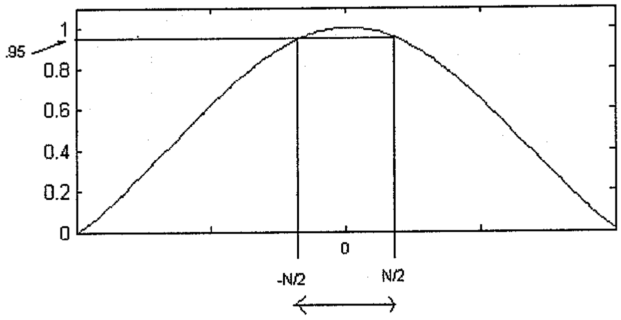


Figure 2-1, Gaussian light wave front; Light amplitude as a function of position; N=number of columns in the data page

2.3 Spatial Light Modulator (SLM)

Following SLM impairments are included in the channel model.

2.3.1 Amplitude contrast ratio (ACR)

The input of SLM is a page of binary data, but in a real SLM, pixels cannot achieve zero intensity. Instead, the zero bits will have a non-zero amplitude leading to a finite Amplitude Contrast Ratio (ACR) ε. ACR is defined as the ratio of the average amplitude of 'one' bits to the average amplitude of 'zero' bits. Thus, in the computer model, all the zeros in SLM plane are replaced by $1/\varepsilon$ (Fig. 2-2). $\varepsilon = 10$ is the default choice.

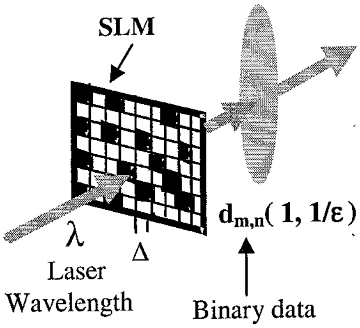


Figure 2-2. Amplitude Contrast ratio

2.3.2 Non-uniformity

The SLM non-uniformity is a measure of the amount of variation in pixel intensities across the SLM. In this simulator, non-uniformity is modeled by a random number, drawn from a uniform distribution. The non-uniformity percentage is an input parameter of the model and controls the range of the uniform distribution. Non-uniformity is a tunable parameter and we use the default value of 1%.

2.3.3Electronic noise

Electronic noise is another source of impairment in the SLM. The SLM electronic noise is modeled as additive white Gaussian noise (AWGN). The mean and variance of AWGN are tunable and this noise is added to the zero-noise SLM values.

2.3.4 Phase mask

The transmittance function of a two dimensional SLM can be described as follows.

$$f(x,y)=\sum_{n=-N/2}^{N/2}\ \sum_{m=-M/2}^{M/2} f(m,n)\,\mathrm{Rect}\left(\frac{x-m\Delta}{\alpha\Delta}\right)\mathrm{Rect}\left(\frac{y-n\Delta}{\alpha\Delta}\right) \qquad (2\text{-}1)$$

Where M is the number of columns in the data page, N is the number of rows in the data page, Δ is the pixel pitch in the SLM and $\alpha$is the SLM linear fill-factor. The 2D array $f(m,n)$ contains the binary data page except that 0 is replaced by $(1/\varepsilon)$.

When the transmittance function $f(x,y)$ of the SLM is illuminated by a plane wave-front, the field in the back focal plane of the lens is the 2D Fourier transform of this function [1]. This Fourier transform has a large dc peak in the recording plane because the input data has zeros and ones and thus has a nonzero DC value. This large DC peak makes the dynamic range in the frequency plane very large leading to potential saturation nonlinearities in the recording medium. One way for making intensity more uniform in the frequency plane is to use a phase mask in the SLM plane. In this method, a mask with randomly varying phase elements is used to multiply the SLM pixel values. The transmittance function, in the presence of a phase mask, is given as follows.

$$f(x,y)=\sum_{n=-N/2}^{N/2}\sum_{m=-M/2}^{M/2} f(m,n)\,e^{j\phi(m,n)}\mathrm{Rect}\left(\frac{x-m\Delta}{\alpha\Delta}\right)\mathrm{Rect}\left(\frac{y-n\Delta}{\alpha\Delta}\right) \qquad (2\text{-}2)$$

$$-\pi\le\phi(m,n)<\pi$$

where $\phi(m,n)$ is the phase element at the (m,n) pixel in the SLM page. Multiplication by a phase mask results in the convolution of the original spectrum by the Fourier transform (FT) of the phase mask. Convolution tends to broaden the peaks, making the dynamic range smaller and thus more acceptable. The phase steps can have values of $2\pi/M$ where M is an integer. Phase mask steps are drawn from a uniform random distribution with M levels in this simulator. The number of phase levels or steps is tunable.

2.3.5 Fill factor

The active portion of an SLM pixel is typically smaller than the entire pixel. The ratio of the active area to total pixel area is called fill factor. In the presence of non-full fill factors, we can rewrite the transmittance function of the SLM as follows.

$$f(x,y)=\left(\sum_{m=-M/2}^{M/2}\sum_{n=-N/2}^{N/2} f(m,n)\,\delta(x-m\Delta,y-n\Delta)\right)*(\mathrm{Rect}\left(\frac{x}{\alpha\Delta}\right)\mathrm{Rect}\left(\frac{y}{\alpha\Delta}\right)) \qquad (2\text{-}3)$$

where * denotes 2D convolution, α is the linear fill factor and $\delta(.)$ is a delta function. The field in the back focal plane of the lens is the Fourier transform (FT) of the transmittance function (in equation 2-3) and is given as follows.

$$F(u,v)=\left(FT\{f(m,n)\}*\sum_{k}\sum_{l}\delta\left(u-\frac{k}{\Delta},v-\frac{l}{\Delta}\right)\right)\text{sinc}\left(u\alpha\Delta\right)\text{sinc}\left(v\alpha\Delta\right) \qquad (2\text{-}4)$$

where Sinc(x) = Sin(πx)/(πx) and where u and v denote the spatial frequencies corresponding to x and y. In the frequency domain, as we can see from equation 2-4, fill factor α is a factor in the sinc function envelope. We simulate the SLM fill factor effect in the frequency domain. The SLM fill factor parameter (α) is tunable and its default value is 0.95.

2.4 Storage media

Storage media impairments are as follows.

2.4.1 Frequency Plane Aperture

An aperture is used in frequency plane and acts as a spatial low-pass filter. By using an aperture in the recording plane, we reduce the amount of media used for storage of that hologram and thus increase the storage density. However, smaller apertures result in more inter-symbol interference (ISI).

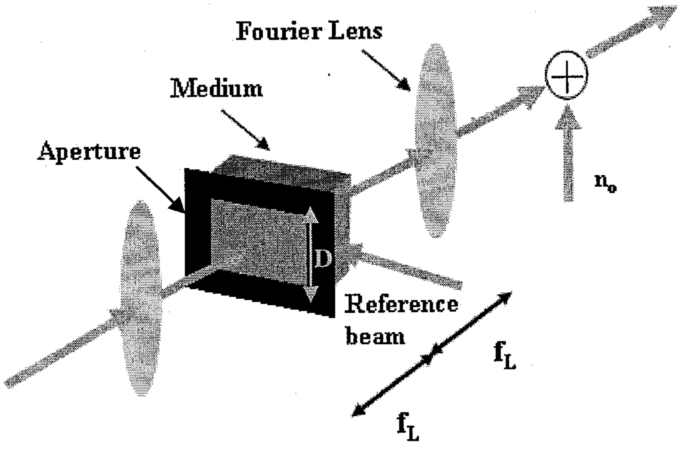


Figure 2-3, Frequency Plane Aperture

An aperture of size $D=\lambda f_L/\Delta$ where Δ is the SLM pixel pitch, λ is wavelength and f_L is the focal length, is known as the Nyquist aperture (Fig. 2-3).

In the real physical system, the aperture width can be bigger than the Nyquist width. To model apertures bigger than Nyquist width, the output of the SLM is up-sampled by factor of two in each direction. This allows us to simulate aperture widths up to twice the Nyquist width. In this simulator, the aperture modeled is a square aperture and is centered at the origin of the frequency plane. The aperture width is a tunable parameter and its default value is 1.21 times the Nyquist area.

2.4.2 Optical noise

During reconstruction, the light amplitude field in the back focal plane of the lens is the Fourier transform of the page recorded in storage media. So the recovered page from the media, after the inverse FT due to the second lens, is back in the spatial domain. To reflect the various noises (e.g., light scatter, lens reflections, stray light, etc.), that affect the light amplitude, we add optical noise. Optical noise is modeled as circularly symmetric complex Gaussian noise as below.

$$n_o = n_i + jn_q \qquad (2\text{-}5)$$

AWGN is added to the real and imaginary parts of the light amplitude just before the camera plane (Fig. 2-3).

2.5 Camera

We consider the following impairments for the camera.

2.5.1 Electronic noise

Camera electronic noise is modeled as AWGN. This noise is added to the amplitude of camera value. The variance and mean are tunable.

2.5.2 Dark noise

Another source of noise in the camera is the dark noise. This is the signal produced by the pixel in the absence of any incident light and is due to thermally generated electrons rather than optically generated electrons. In our simulator, dark noise is modeled by a uniform distribution. The dark noise intensity percentage is a tunable parameter.

2.5.3 Camera fill factor

Just like SLM fill factor, the camera fill factor is the ratio of active area to the total pixel area. We simulate the effect of camera fill factor in the Fourier transform of the amplitude. After adding optical noise to the light wave front impinging the camera, Fourier transform is applied to the noisy wave front to go to frequency domain. Sinc function equations are same as in the case of SLM fill factors (equations. 2-1, 2-3 and 2-4). An inverse FT is applied to determine the effective light amplitude wave front at the camera. After applying fill factor in frequency domain and coming back to spatial domain, intensity incident on camera is determined. Any fill factor value is possible in this method, but we need to perform two Fourier transforms. Areal Camera fill-factor is a tunable parameter and the default value used is 0.40.

We can apply camera fill factor directly in camera plane. For this method first, we need to over sample the pre-camera page. Then set a select number of sub-pixels in each pixel to zero to simulate different detector fill factors. The ratio of the number of non-zero intensity sub-pixels to all of the sub-pixels is equal to camera fill factor. Finally calculate intensity over each pixel. This method has two disadvantages. First, over sampling increases the computational complexity. Second, we can apply limited value of fill factor by finite over sampling. As a result we didn't use this method.

2.5.4 Quantization

Another type of camera impairment is the quantization error. In this simulation, we consider non-uniform quantization with fixed upper and lower limits of 0 and 1, respectively. Based on the real data provided by Inphase Technologies, it is evident that the output levels are not quantized uniformly; they are quantized based on a seemingly consistent non-uniform quantizer function. For instance, we have determined from observation that the input-output curve of the quantizer for a set of 64 consecutive output levels out of the 1024 levels is as in Figure 2-4.

Quantization level is a tunable parameter as a power of two (number of bits).

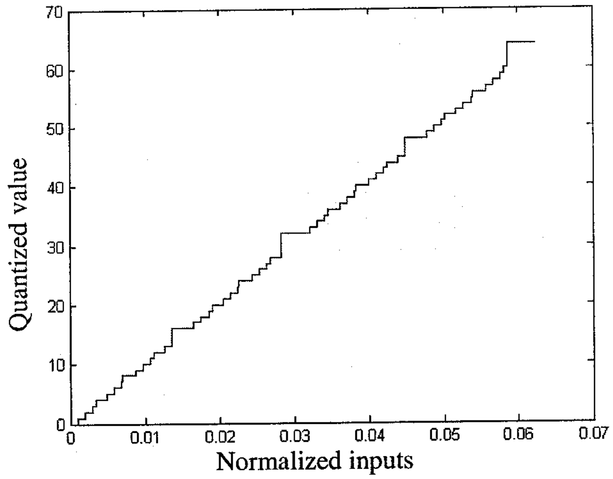
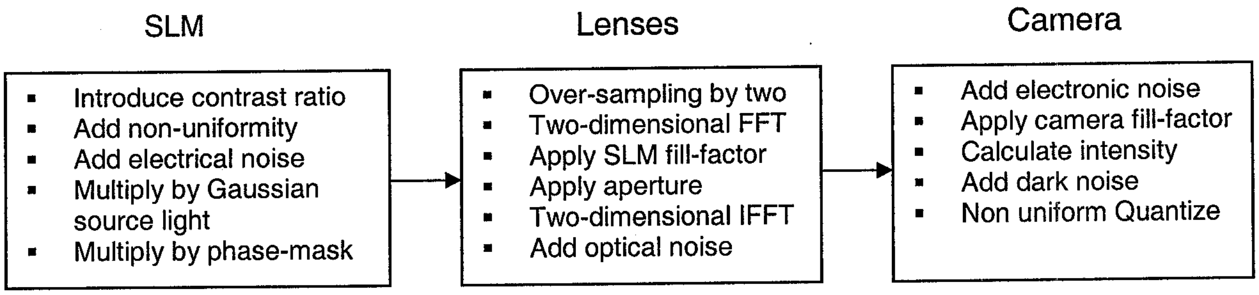


Figure2-4, Input-output curve for the quantizer for one set of 64 output levels

2.6 Overall simulation

Overall simulation can be summarized as follows:



2.7 Summary

Tunable channel model has an important role in investigating signal processing options. As a result, physical channel model with tunable parameters is developed. This simulator is implemented based on mathematical equations of channel impairments.

Chapter 3

Evaluation of Physical Channel Model and Channel Identification

In this chapter we first evaluate the performance of the physical channel model using the normalized root mean squared error (NRMSE) and the signal to noise ratio (SNR) metrics. We used 60 input pages and corresponding 60 recovered output pages provided by Inphase Technologies. For NRMSE metric calculations, we compare the simulated output page with real recovered output page pixel by pixel and for SNR metric, we compare the histogram of simulated output pages with histograms of real output pages.

Then we investigate the 2D linear and 2D non-linear channel identification methods using these 60 input pages and corresponding output pages as training pages. We used linear minimum mean square estimation as the linear method and look-up-table as the nonlinear method. We also compare their performance with that of physical channel model.

3.1 Evaluation of physical channel model

To evaluate the validity of the physical channel model we compare the outputs of the model with the real output pages using the same input pages. We used 60 real output pages and corresponding input pages provided by Inphase Technologies. The size of data pages is 1024X1280 with binary input data. The parameters used in simulator for these result are:

- Amplitude contrast ratio = 10
- Nonuniformity = 1%
- Light drop in the edges = 10%
- Number of Phase mask levels = 16
- Areal SLM fill factor = 95%

- Areal aperture size = 1.21×Nyquist aperture
- Areal camera fill factor = 40%
- Optical noise: mean = 0, variance = 0.023
- Electronic noise: mean = 0.42, variance = 1e-5
- Dark noise: 1%
- Non-uniform quantizer : 10 bit with upper limit = 1024 and lower limit =0

3.1.1 Histogram comparison

The output page histograms of the zeros and ones give an idea of how well they are separated. In Figures 3-1 the histograms of a real page and simulated page generated with the same input pattern are shown and appear to be quite similar. Default parameters are used for simulator.

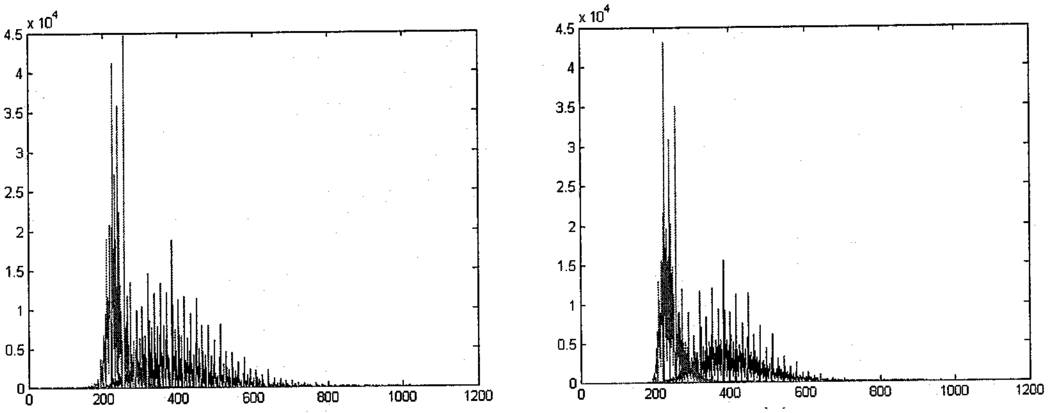


Figure 3-1, Histograms of (a) Recovered and (b) Simulated pages

The separation between the histograms can be quantitatively represented using the *SNR* metric defined below

$$SNR_{page} = 20\log_{10}\left(\sqrt{2}\ erfc^{-1}\left(2\ BER_{page}\right)\right) \qquad (3\text{-}1)$$

BER_{page} is the average of theoretical Bit Error Rates (*BER*) over several tiles (sub-blocks) of the input page. BER of each tile is obtained using equation 3-2.

$$BER = \tfrac{1}{2} erfc\left(\frac{Q}{\sqrt{2}}\right) \qquad (3\text{-}2)$$

Q is given by:

$$Q = \frac{\mu_1 - \mu_0}{\sigma_1 + \sigma_0} \qquad (3\text{-}3)$$

Where μ_0 and $\mu 1$ are the means of intensities of camera pixels corresponding to zeros and ones respectively and σ_0 and σ_1 are standard deviation of intensities of camera pixels corresponding to zeros and ones, respectively, in the input page.

Typical SNR obtained using tiles of size 32x32 pixels in the real page was about 2.8dB and in simulated pages of the same tile size about 3.3dB as can be seen in Fig. 3-2. InPhase Technologies has been able decode without errors encoded pages with SNR values around 2.8dB.

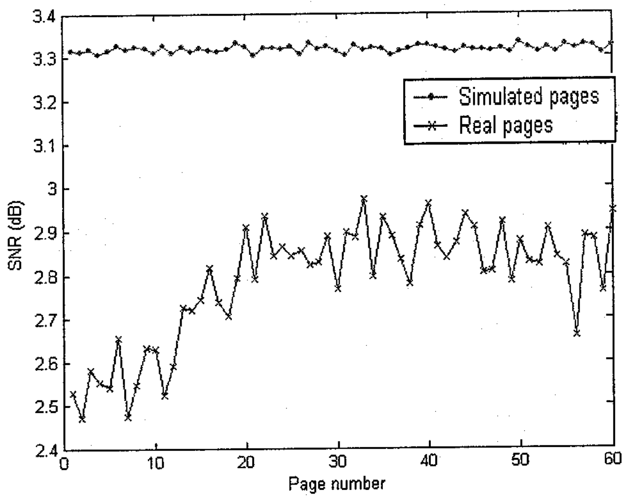


Figure 3-2, SNR of Real and simulated pages

3.1.2 Page comparison

The level of closeness between the pixels in the real and simulated pages is quantified by a metric called the Normalized Root Mean Square Error (NRMSE), defined as follows.

$$NRMSE = \sqrt{\frac{\left|\sum_{m,n} |R(m,n) - \alpha S(m,n)|^2\right|}{\sum_{m,n} R(m,n)^2}} \qquad (3\text{-}4)$$

Where R is the recovered page and S is the simulated page and m and n are the rows and columns of the pixels. The α that minimizes the numerator of equation 3-4 is given below in equation 3-5.

$$\alpha_{\min} = \frac{\sum_{m,n} R(m,n)S(m,n)}{\sum_{m,n} S(m,n)^2} \quad . \qquad (3\text{-}5)$$

The NRMSE between the simulated and real pages ranges between 29-32%. A plot of this is shown in Figure 3-3 (a). Each point represents the NRMSE of one page. The corresponding α values for the NRMSE values are plotted in Figure 3-3 (b). From the α values we infer that they are close to 1 for all the pages, which means that the simulated pages do not require an additional scaling factor and the α values do not vary much for the different pages.

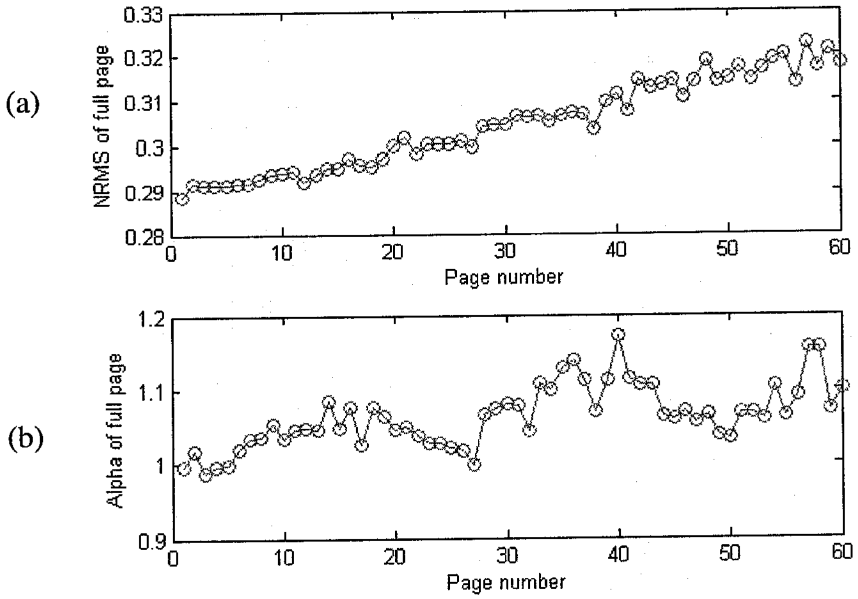


Figure 3-3, (a) NRMSE between recovered and simulated pages, (b) Corresponding α values.

We see that some portions of simulated pages are more accurately simulated than others. The white regions in Figure 3-4 have NRMSE less than 28% and there are about 64% such blocks.

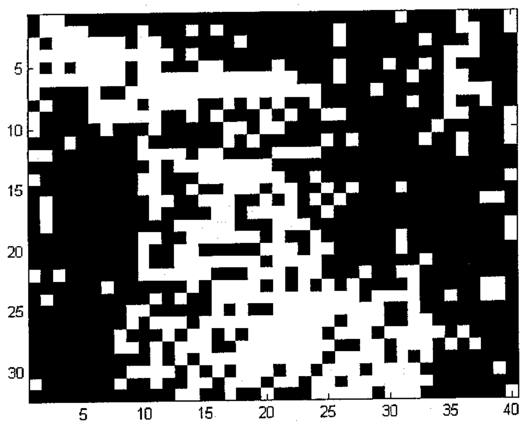


Figure 3-4, Black blocks (sized 32×32 pixels) are regions in a page having NRMSE between real and simulated pages ≤ 28%

3.2 Channel identification

In this section, we report the results of our investigation of Linear Minimum Mean Square Error estimation and nonlinear look-up-table methods. The 60 input binary pages and corresponding real pages have been used as training pages. The performances of these methods are compared using the normalized root mean squared error (NRMSE) metric.

3.2.1 Linear MMSE channel identification

As we explained earlier, for holographic data storage channel there are many sources of impairment and noise that reduce the fidelity of output data. To obtain a simple model, it is useful to consider two main factors. The first factor is concerned with the imaging properties of the optical system. HDS optical system has a low-pass response that results in 2D blurring. That optical blur causes 2D ISI in the recovered pages. Another important factor is the noise. HDS systems consist of many supporting devices and systems and each of them contributes its own noise. To model the channel we consider a Gaussian noise model as the overall noise of the HDS. So we can consider HDS as a 2D ISI channel whose output is corrupted by additive white Gaussian noise (AWGN) as shown in Fig. 3-5. The mathematical expression of this channel is as in equation 3-6.

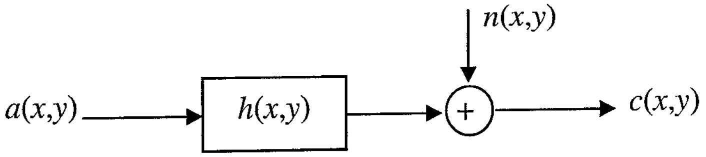


Figure 3-5, Schematic of linear AWGN channel

$$c(x,y)=\sum_{p=-L}^{L}\sum_{q=-L}^{L}h(p,q)a(x-p,y-q)+K+n(x,y) \qquad (3\text{-}6)$$

where *c(x,y)* is the simulated output, *a(x,y)* is the input data page, *L* is the number of pixel in each direction that ISI extends and *n(x,y)* is the Gaussian noise. Constant *K* is used to model the effect of nonzero means of input and output pages. To calculate the coefficients of *h(x,y)* we have used the Linear Minimum Mean Square Error (LMMSE) estimation method given by the equation 3-7.

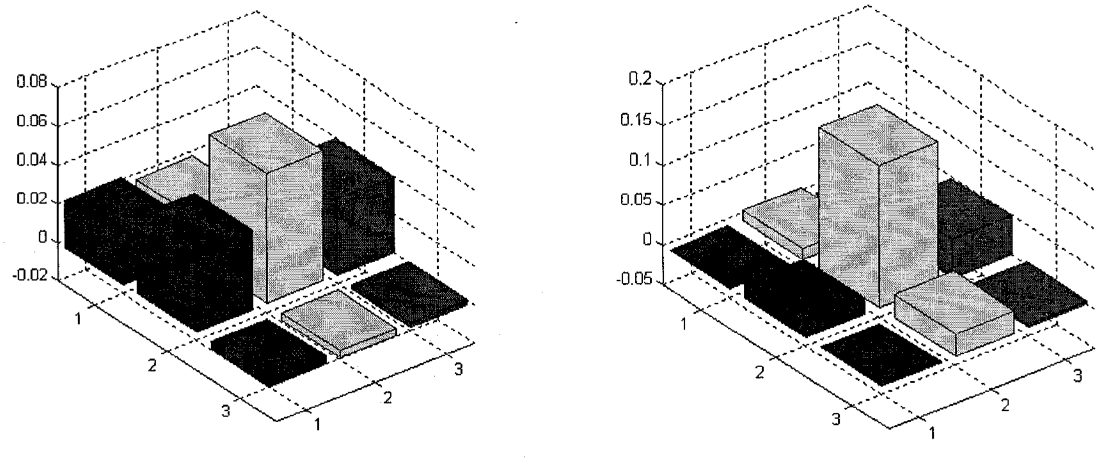
$$C_{AA}\,h = C_{AC} \qquad (3\text{-}7)$$

Where C_{AA} is the input autocorrelation and C_{AC} is the input-output cross correlation. C_{AA} and C_{AC} are computed using real recovered output page and corresponding input page.

We have considered *h(x,y)* as a 3×3 kernel. In another word the ISI is truncated just to one pixel in each direction (*L*=1). Because of the non-stationary nature of output pages (i.e., optical properties are better along the optical axis than away from it), we determine channel model over each 64×64 block of a page. In Fig. 3-6 the channel model parameters of corner block and middle block of page one is shown. The channel model parameters for all pages are very close to those of page one. We can see from Figure 3-6 that the ISI for corner block is more than that of the center block, but in general the amount of ISI is not much.

The ISI can be quantitatively represented by equation 3-8.

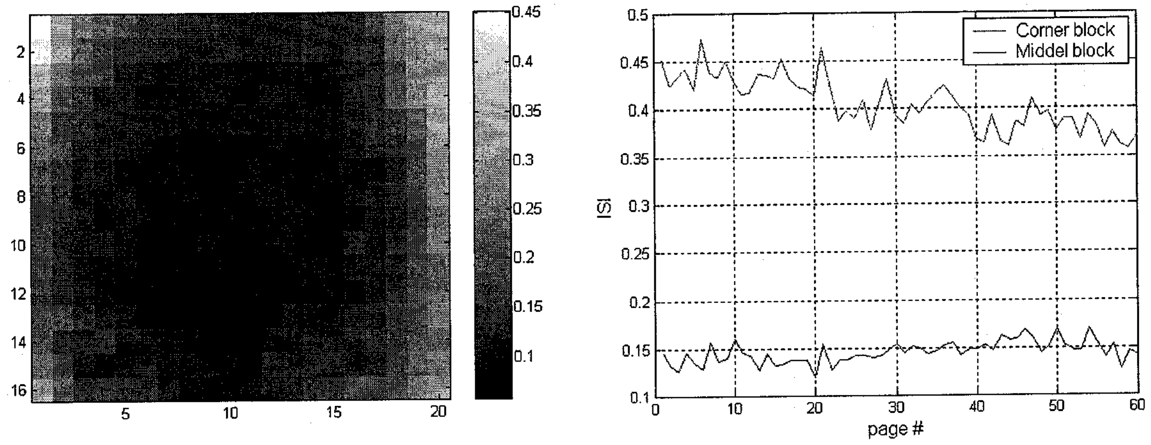
$$ISI=\frac{\sum h(i,j)^2-h(2,2)^2}{\sum h(i,j)^2} \qquad (3\text{-}8)$$



(a) **(b)**

Figure 3-6, Channel model of 64×64 block (a) corner block (b) center block

Fig. 3-7 (a) shows the ISI for all the 64×64 blocks of page one and Fig. 3-7 (b) shows the ISI of corner blocks and middle blocks of all 60 pages.



(a) **(b)**

Figure 3-7, ISI of (a) all 64×64 blocks of page one, (b) center and middle blocks ISI of 60 pages

With this linear channel model, we can estimate the noise also. Noise is the difference between the channel model output and real output. The histogram of noise for page one is shown in Fig. 3-8. As we can see the distribution of error is not exactly Gaussian but close to it. If modeled as

Gaussian, the variance of error is 0.0033 and the mean is almost zero. The root mean square error between channel model output and real output for page one is 0.177.

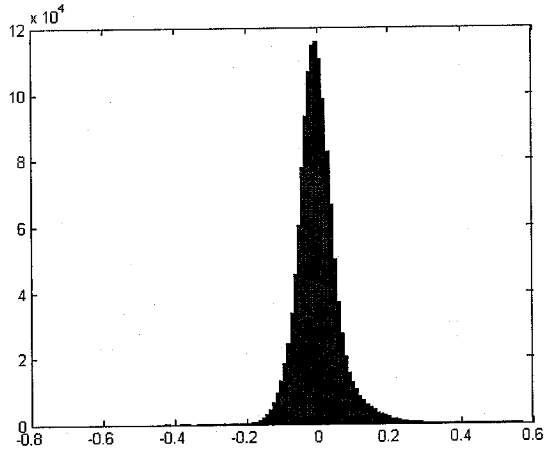


Figure 3-8, Histogram of identification error for page 1

3.2.2 Non-linear Look-up-table channel identification

Look-up-table method is a non-linear channel model. To build the table we consider blocks of 3×3 in input pages. These 3×3 blocks contain all the possible combinations of ones and zero therefore the look-up-table has 512 entries as shown in Fig. 3-9. For each entry of the table, a number representing the output value is needed. Output values are the average of center points of the 3×3 block in the real output pages for the corresponding input blocks. For example for the first entry of the table (all zero 3×3 block) we look for all-zero 3×3 blocks in the input page and find the center values of corresponding 3×3 blocks in output pages and find their average. This value is the output of the first entry of table (C_0 in Fig. 3-9). An output page can be simulated by using the table. Because of the non-stationarity of real output pages, each 64×64 block of a page has its own table. Although, this method is computationally demanding, it captures nonlinearities and may provide improved results.

$$\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}\longrightarrow C_0 \qquad \begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}\longrightarrow C_1 \qquad \begin{bmatrix}0&0&0\\0&0&0\\0&1&0\end{bmatrix}\longrightarrow C_2$$

$$\begin{bmatrix}0&0&0\\0&0&0\\0&1&1\end{bmatrix}\longrightarrow C_3 \qquad \begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix}\longrightarrow C_4 \qquad \begin{bmatrix}0&0&0\\0&0&0\\1&0&1\end{bmatrix}\longrightarrow C_5$$

$$\begin{bmatrix}1&1&1\\1&1&1\\1&0&1\end{bmatrix}\longrightarrow C_{509} \qquad \begin{bmatrix}1&1&1\\1&1&1\\1&1&0\end{bmatrix}\longrightarrow C_{510} \qquad \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\longrightarrow C_{511}$$

Figure 3-9, Look-up-tables. Each 3x3 blocks of ones and zero is an entry of look-up-table and each *C* *is* corresponding output

3.3 Channel estimators result

To compare the performance of the LMMSE channel identification and the nonlinear look-up-table method, the NRMSE metric in equation 3-4 is used. The result is shown in Figure 3-10. In this figure the NRMSE of physical channel model is also plotted. The performance of look-up-table is better than those of the LMMSE identification and the physical channel model.

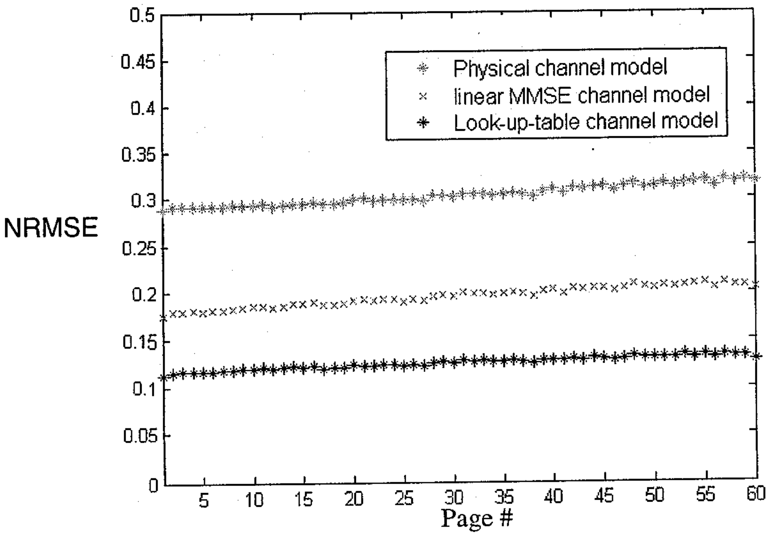


Figure 3-10, comparison of NRMSE for LMMSE channel identification, look-up-table and physical channel model

3.4 Summary

The LMMSE and the look-up-table channel models both outperform the physical channel model. But physical channel model has some advantages over the LMMSE and the look-up-table models. One advantage of physical channel model is its capability of tuning the parameters of the simulator. Therefore we can investigate the channel by changing its physical parameters. The nonlinearity of the channel is included in this model too. Even though the NRMSE of physical channel model is more than those of other models, for some parts of a page the NRMSE is very low. While the performance of look-up-table is the best, it is not computationally practical.

Based on these channel models, we can infer that the holographic data storage channel is a low SNR channel and that the ISI in central part of a page is low but is more in the outer regions.

Chapter 4

2D Equalization for Holographic Data Storage Channel

Holographic data storage (HDS) channel suffers from 2D Inter-Symbol Interference (ISI) and a standard approach to deal with ISI is the use of equalization. In this chapter, we investigate linear minimum mean squared error (MMSE) and zero-forcing (ZF) equalization methods and nonlinear adaptive Volterra and adaptive decision feedback (DF) equalization methods while considering 2D nature of HDS. The performance and complexity of these equalizers are compared using 60 real data pages supplied by InPhase Technologies as well as simulated data. We have used the physical channel model for generating simulated data.

In Fig. 4.1 the schematic of a system with equalizer is shown where *Y* represents the output page (equalizer input) and *R* is the equalized page (equalizer output). *A*(*m,n*) denotes the digital (usually binary) data being stored. The output page can be modeled (assuming linearity) as the 2D convolution of the *A*(*m,n*) with the impulse response of the channel, except for the additive noise.

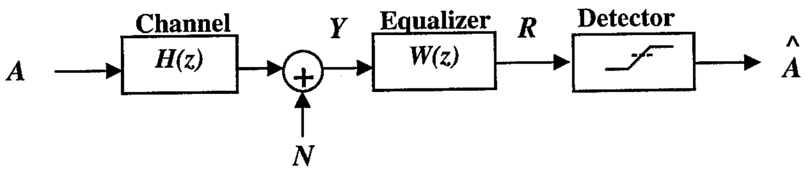


Figure 4.1, A schematic of equalization

Since the channel impulse response is usually not a delta function, the channel suffers from ISI and simple detection schemes will result in significant errors if the ISI is left untreated.

4.1 MMSE Equalizer

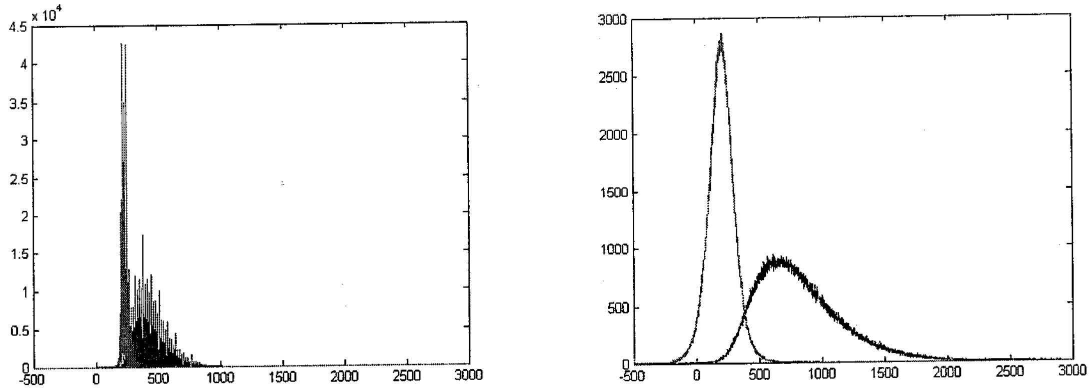
MMSE equalizer is defined by a (2L+1)x(2L+1)+1 array of coefficients *w*. Its operation is as in equation 4-1.

$$R(m,n) = \sum_{p=-L}^{L} \sum_{q=-L}^{L} w(p,q).y(m-p,n-q) + w_0 \qquad (4\text{-}1)$$

The w_0 coefficient allows us to model the nonzero mean of recovered data pages (Y). Coefficients w are obtained by

$$C_{YY}\, w = C_{YA} \tag{4-2}$$

where C_{YY} is the output autocorrelation and C_{YA} is the input-output cross correlation. The w coefficients are selected to minimize the mean squared error between the equalizer output and a target output. The target output here is the input page. Figure 4.2 shows the histograms before and after MMSE equalization. It is clear that the histograms have less overlap after equalization.



(a) **(b)**

Figure 4-2, Histograms of ones and zeros before (a) and after (b) minimum mean squared error (MMSE) equalization

4.2 Zero forcing equalizer (ZFE)

A straightforward equalizer is the zero-forcing equalizer (ZFE), which essentially inverts the transfer function $H(z)$ of the channel, so that the overall transfer function $H(z)W(z)$ is 1 for all frequencies, thus having no ISI. However, $H(z)$ is typically a low-pass type of channel causing $W(z)$ to be large at high frequencies. So the disadvantage of this equalizer is that it amplifies high frequency noise causing the BER to increase. The ZF equalizer is obtained from equations 4-3 and 4-4.

$$R(n,m) - m_R = w * (Y(n,m) - m_Y) \tag{4-3}$$

$$FFT(R) = FFT(Y).FFT(w) + w_0\delta(z) \tag{4-4}$$

4.3 Adaptive decision feedback equalizer

The basic idea of decision feedback equalization (DFE) is to pass the read-back signal through a linear equalizer called the forward equalizer, so that the non-casual or the precursor ISI is made zero. Previous decisions at the output of the threshold detector are fed back through the backward equalizer, which eliminates the post-cursor or casual ISI left after the forward equalizer. A schematic diagram of the adaptive DFE is shown in Figure 4-3.

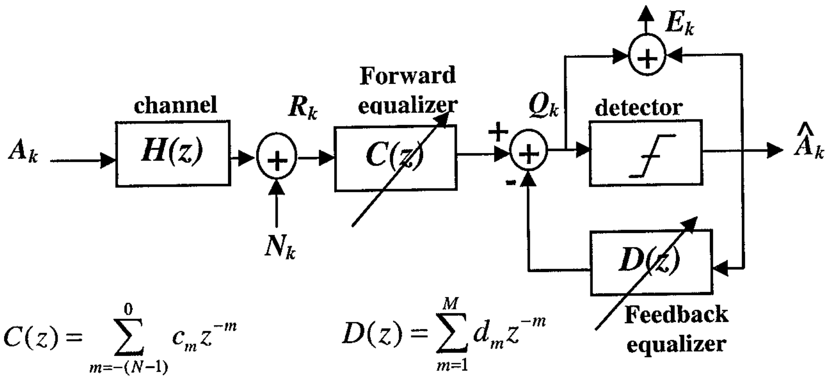


Figure 4-3, A schematic of adaptive DFE

Since the forward equalizers are designed to minimize the noise variance and they allow post-cursor ISI, DFE has less noise enhancement compared to linear equalizers that try to cancel all ISI. However, the big disadvantage of DFE is the error propagation due to potentially incorrect decisions made by threshold detector, particularly in low SNR. This error propagation will be a big problem in low SNR channels.

Equalizer weights are updated adaptively according to the steepest descent algorithm using following equations (equations 4-5 to 4-10) when considering the2D nature of the system.

$$E_k = Q_k - \hat{A}_k \tag{4-5}$$

$$Q_k = \sum_{i=-(N-1)}^{0} c_i R_{k-i} - \sum_{i=1}^{M} d_i \hat{A}_{k-i} \tag{4-6}$$

$$V^T = [c_{-(N-1)} \cdots c_0 \; d_1 \cdots d_m] \tag{4-7}$$

$$W_k^T = [R_{K+(N-1)} \ldots R_k \; A_{k-1} \ldots A_{k-M}] \tag{4-8}$$

$$E_k = \hat{A}_k - V_k^T W_k \tag{4-9}$$

$$V_{k+1} = V_k + \beta E_k W_k \tag{4-10}$$

where E_k is the error signal between input and output of the threshold detector (Fig. 4-3). $\hat{A}_k$ is detected data (the assumption is $\hat{A}_k = A_k$) and Q_k is reduced ISI signal. E_k can be written as equation 4-9 too. V is an array containing coefficients of forward equalizer and feedback equalizer (equation 4-7). W_k is an array containing the inputs of forward equalizer and feedback equalizer with assumption that $A_k = \hat{A}_k$ (equation 4-8). k is iteration number and β is updating step. Equation 4-10 is the updating equation for the equalizer's coefficients

4.4 Adaptive Volterra Equalizer

The quadratic relationship between input data and intensity output of camera suggests the use of a nonlinear equalizer modeled as a second order discrete time Volterra equation. The 2D equalizer output is obtained using equation 4-11.

$$\begin{aligned} R(m,n) = w_0 + \sum_{p_1=-N}^{N} \sum_{p_2=-N}^{N} w_1(p_1,p_2).y(m-p_1,n-p_2) + \\ \sum_{p_1=-M}^{M} \sum_{p_2=-M}^{M} \sum_{p_3=-M}^{M} \sum_{p_4=-M}^{M} w_2(p_1,p_2,p_3,p_4).y(m-p_1,n-p_2).y(m-p_3,n-p_4) \end{aligned} \tag{4-11}$$

where $y(m,n)$ represents the recovered page (equalizer input), $\hat{Q}$ is the equalized page (equalizer output) and w_0, w_1, w_2 are equalizer constants and kernels. The w_1 is a 2D kernel of size $(2N+1)$ by $(2N+1)$ whereas w_2 is a 4D kernel of size $(2M+1)$ along each axis. The weights of equalizer are adapted according to the least mean square error (LMS) algorithm.

$$\begin{cases} w_{0_{k+1}} = w_{0_k} + \alpha e_k \\ w_{1_{k+1}} = w_{1_k} + \beta e_k Y(n,m) \\ w_{2_{k+1}} = w_{2_k} + \gamma e_k U(n,m) \end{cases} \tag{4-12}$$

where k is iteration number, α, β and γ are updating steps, Y is a vector containing $y(m,n)$, U is a vector containing the products of $y(m,n)$ (U:{ $y(m-p_1,n-p_2)$ X $y(m-p_3,,n-p_4)$ }), and e_k is the error signal.

4.5 Results with real data and simulated data

We have applied the above equalizers to 60 real un-coded data pages of size 1024×1280. Because of the non-stationary nature of the underlying ISI, we process blocks of size 64X64. For MMSE, ZF and adaptive DF equalizers, a 3×3 kernel with a constant w_0 is considered. A bigger kernel did not result in a noticeable improvement. For adaptive Volterra equalizer we use a 3×3 kernel for w_1 and a 3×3×3×3 kernel for w_2 and a constant w_0. The resulting BERs are shown in Fig. 4-4. The SNR of these recovered pages is very low (around 3 db) so the BER is very high. These BERs are for un-coded data. The BER improvement for MMSE equalizer is about 20%. The performance of ZF algorithm is a little worse than that of the MMSE. The performance of adaptive Volterra equalizer is very close to that of the MMSE. Adaptive DF equalizer performance is worse than that of the MMSE because of its error propagation.

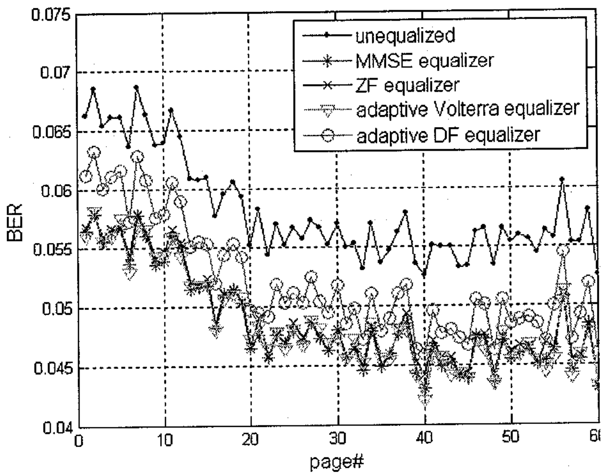


Figure 4-4, Equalizers for real data

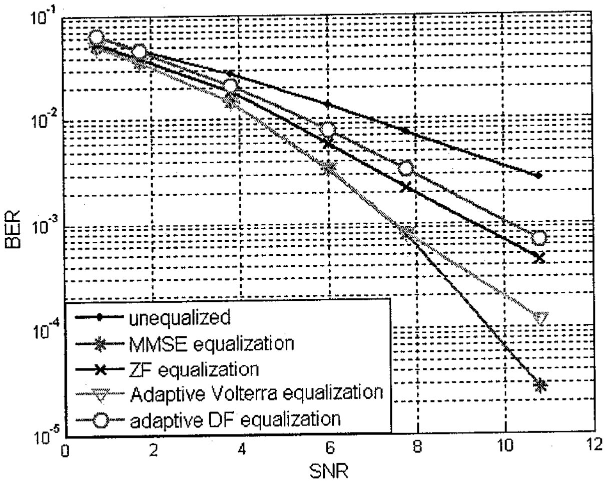


Figure 4 -5, Equalizers for real data

To investigate the performance of these equalizers for higher SNR data pages, we used the physical channel model. In our investigations, MMSE equalizer shows better performance compared to others for higher SNR (Fig.4-5). The SNR calculated based on equation 4-13 where E_s is normalized energy of signal and σ^2 is the variance of noise.

$$SNR = 20\log\sqrt{\frac{E_s}{\sigma^2}} \qquad (4\text{-}13)$$

4.6 Practical implementation of MMSE equalizer

Because MMSE equalizer shows better performance, we focus on MMSE equalizer for practical implementation. In experiments described so far, we determine the MMSE equalizer coefficients by using all the data of input and output pages. This method is computationally demanding. Also in a real system we don't know most of the input data on a page (except for some fiducials) since it is user data and thus cannot hope to use all the bits on a page to determine equalizer coefficient. We tried two more practical variations. The first is using one page as a training page and applying the resulting equalizer to other pages. Second uses a training page to obtain equalizer coefficients and updates them for each page based on a small part of the block (e.g., a 8×8 sub-block where known bits are recorded). Both are promising as shown in Fig. 4-6.

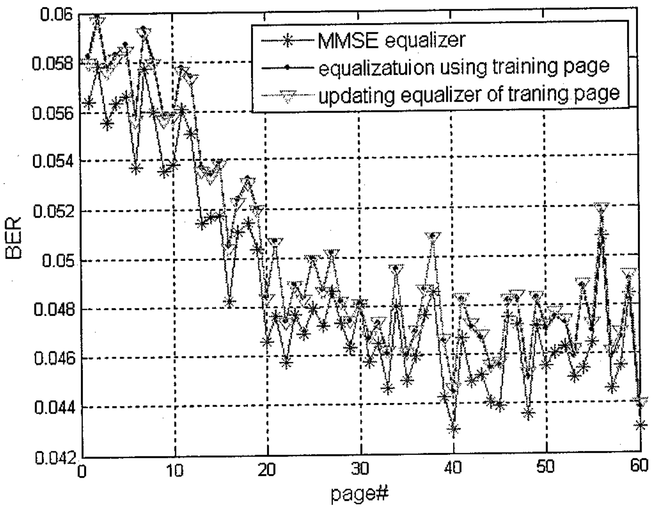
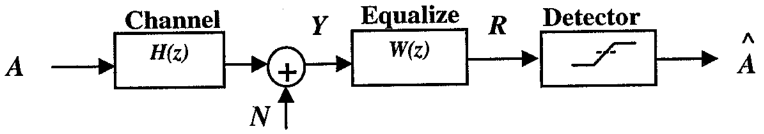


Figure 4-6, MMSE equalizer obtained from training page and MMSE equalizer updating for each page

4.7 Simplified implementation of MMSE equalizer

One method to compute the ZF equalizer coefficients is in frequency domain. In this method, to have a good result, we need to use all the pixels of an input page and corresponding real output page to calculate the coefficients of the ZF equalizer. We also need Fourier and inverse Fourier transform. Adaptive Volterra equalizer, beside 2D kernel, has 4D kernel (w_2) and these are estimated iteratively. In DF equalizer, we have two 2D kernels that are obtained by iteration. So the MMSE equalizer besides offering better BER performance, is less complex than the other equalizers. We have also designed simplified MMSE equalizers by requiring symmetric and separable kernels with fixed 8-point parameters. Symmetric kernel reduces the number of multiplications. For symmetric channel we need to find 4 parameters instead of 9 parameters so we can add the values that are multiplied by the same coefficient and carry out just one multiplication. By using a separable equalizer kernel instead of 2D convolution, we can use two 1D convolutions and we need to find only 2 parameters. Schematic of MMSE equalizer, symmetric and separable kernel are depicted in Fig. 4-7



$$R(n,m) = \sum_{p=-1}^{1} \sum_{q=-1}^{1} w(p,q).y(n-p,m-q) + w_0$$

$$w = \begin{bmatrix} c & b & c \\ b & a & b \\ c & b & c \end{bmatrix}$$

Symmetric 3×3 equalizer

$$w = \begin{bmatrix} b \\ a \\ b \end{bmatrix} \begin{bmatrix} b & a & b \end{bmatrix} = \begin{bmatrix} b^2 & ab & b^2 \\ ab & a^2 & ab \\ b^2 & ab & b^2 \end{bmatrix}$$

Separable 3×3 equalizer

Figure 4-7, Schematic of MMSE equalizer with Symmetric and Separable kernel

The BER results obtained from simplified MMSE equalizer based on symmetric and separable equalizer kernel for real pages and simulated pages are shown in Fig. 4-8 and Fig. 4-9, respectively.

Simulated data were generated by the physical channel model. For low SNR real data, we observe around 15% improvement in BER. Based on these results, simplifying MMSE equalizer does not degrade its performance much and we can reduce computational complexity.

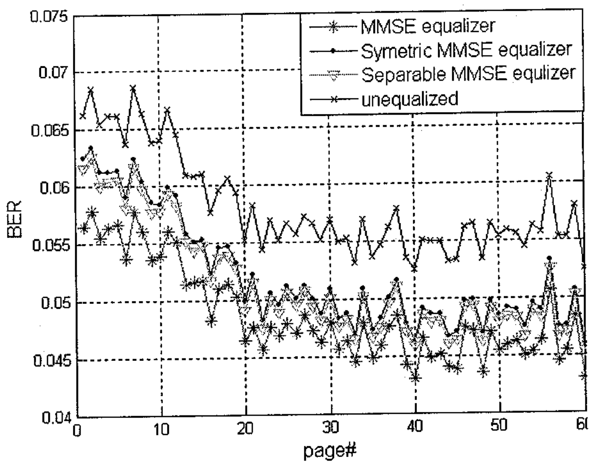


Figure 4-8, BERs for Symmetric and separable MMSE equalizer for real data

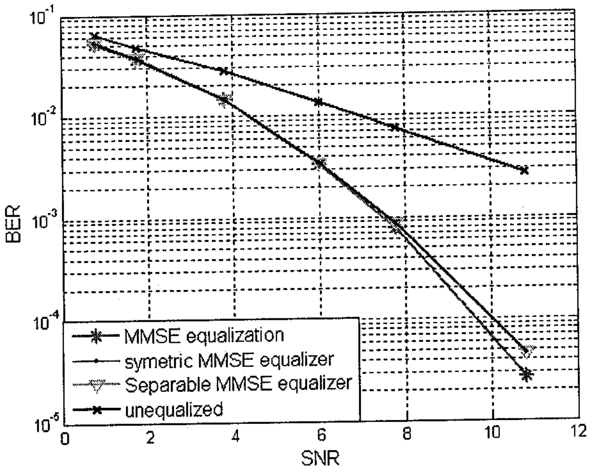


Figure 4-9, BERs for Symmetric and separable MMSE equalizer for simulated data

4.8 Summary

Among the various equalizers investigated (MMSE, ZF, adaptive DF and adaptive Volterra equalizers), MMSE equalizer shows the best performance for real data (with low SNR) and simulated data. The complexity of MMSE equalizer is less than those of other equalizers. Another advantage of MMSE equalizer is that practical and simplified implementations of MMSE equalizer appear to not degrade its performance too much.

Chapter 5

Detection for Holographic Data Storage Channel

In this chapter, the performances of different detectors for holographic data storage channel are compared. We have considered fixed threshold detector, adaptive threshold detector, log likelihood ratio detector (LLR) and iterative detector. Fixed threshold detector is common because of its simplicity. Its performance is good when the amount of noise and ISI is low. Adaptive threshold is useful to assess the performance of other single-bit detectors. LLR detector makes a decision in a sense of maximum likelihood. Maximum likelihood is an optimal detector when the input symbols are equally likely. LLR detector is commonly used with decoder and soft decision detector. Iterative detector takes into account the effects of neighboring bits and 2D ISI. As a result it is a good candidate for HDS. These detectors are applied to real recovered and simulated data as well.

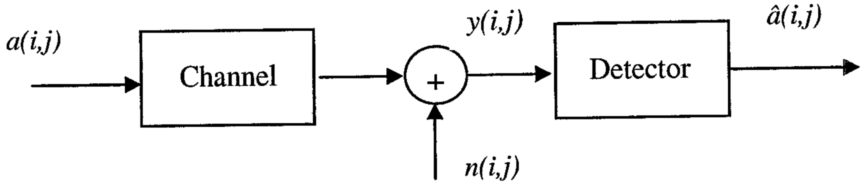


Figure 5-1, Schematic of a channel with detector

5.1 Threshold detector

For threshold detector, a fixed threshold is used. This fixed threshold is obtained by calculating the mean of the output data. Fixed threshold detector is the worst detector for both real data (with low SNR) and simulated data (with higher SNR). Simplicity is the advantage of this detector.

5.2 Adaptive threshold

In adaptive threshold method, we search for a threshold that minimizes the empirical bit error rate (BER). In this method first the mean of output values corresponding to zero input bits and the mean of output values corresponding to ones input bits are calculated. Then between these two means, we search for a threshold that gives the lowest BER on the training data. Although the performance of this detector is very good especially for low SNR real data, it is not a practical approach because the input data has to be known in order to choose the best threshold. In addition, an exhaustive search has to be done. This adaptive threshold detector provides a good benchmark to assess the performance of other single-bit detectors.

5.3 Log Likelihood Ratio (LLR) detector

This detector is designed based on the LLR shown in equation 5-1.

$$LLR = \log\left(\frac{prob.(y\,(i\,,j\,)|a(i\,,j\,)=1)}{prob.(y\,(i\,,j\,)|a(i\,,j\,)=0)}\right) \tag{5-1}$$

Where $y(i,j)$ is the output value and $a(i,j)$ is the corresponding input bit. The decision should be one if the likelihood, given the input bit = one, is more than the likelihood, given the input bit = zero (equation 5-2).

$$\begin{cases} \hat{a}(i,j)=1 & if \quad LLR \geq 0 \\ \hat{a}(i,j)=0 & if \quad LLR < 0 \end{cases} \tag{5-2}$$

The HDS channel suffers from different kinds of noises, but we lump all of them into a single Gaussian noise model. To simplify the LLR calculation, we model the channel as an additive white Gaussian noise (AWGN) channel. From the ones and zeros histograms of output pages shown in Fig 5-2, we can see that the parameters of Gaussian distributions for output bits corresponding to ones and zeros are different.

A practical way to obtain the parameters (means and variances) of Gaussian distributions for ones and zeros is first using fixed threshold and estimating ones and zeros, and using the estimated bits to obtain refined histograms of ones and zeros, and finally calculating the means and variances of ones and zeros. By considering Gaussian distribution, LLR can be calculated easily by equation 5-3.

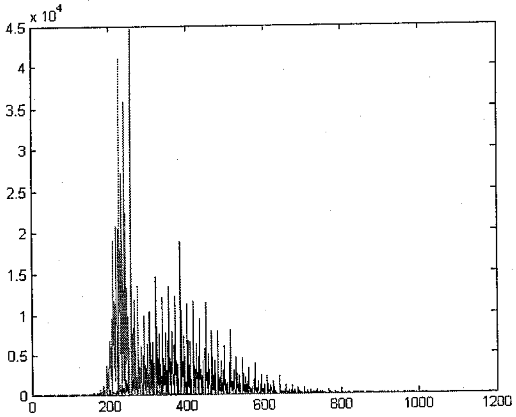


Figure 5-2, Histogram of real recovered data,Red lines for zeros and blue lines for ones

$$LLR = \log\left[\frac{\frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left\{-\frac{(y-m_1)^2}{2\sigma_1^2}\right\}}{\frac{1}{\sqrt{2\pi\sigma_0^2}}\exp\left\{-\frac{(y-m_0)^2}{2\sigma_0^2}\right\}}\right] \qquad (5\text{-}3)$$

where m_1 and m_0 are the means of output bits corresponding to input ones and zeros and σ_1 and σ_0 are the standard deviation of output bits corresponding to input ones and zeros.

The performance of LLR detector for low SNR real data is better than that from the fixed threshold detector, but worse than that of the adaptive threshold scheme. This may be because of Gaussian assumption and using fixed threshold to obtain the parameters of these Gaussian distributions. We can see from Fig. 5-2 that the tails of ones and zeros histograms are not symmetric. Therefore using Gaussian model creates some error. As we said, fixed threshold does not perform well and as a result using fixed threshold to estimate the Gaussian parameter increases the error even more.

5.4 Iterative detection

The previous detectors make decision on each individual bit without taking into account the effects of neighboring bits. But there is 2D ISI and for each bit there are 8 closest neighbors. One way to consider the effects of neighboring bits is using 2D iterative detection motivated by decision feedback techniques. In this detector we need to know the channel model. For simplicity we model the channel as a linear discrete 2D ISI channel whose output is corrupted by Gaussian noise. The mathematical representation of the channel is given by equation 5-4.

$$y(i,j) = \sum_{p=-L}^{L} \sum_{q=-L}^{L} h(p,q)a(i-p,j-q)+n(i,j) \qquad (5\text{-}4)$$

where L is the number of bits in each direction that the ISI extends, $a(i,j)$ is the mean-subtracted input data and $y(i,j)$ is the mean-subtracted output data.

Channel model is calculated by linear minimum mean square error (LMMSE) estimation method, using input pages and corresponding output pages. We obtain channel models for real holographic data storage channel as well as for the simulator channel.

The first step of this 2D iterative detection algorithm is to apply the fixed threshold to the output page, producing an input data estimate $\hat{a}(i,j)$. At each pixel on the page a test is performed based on these estimated data. To test the estimated data $\hat{a}(i,j)$, first we set $\hat{a}(i,j)=1$ and the neighbors as determined by estimation and convolve them with channel model to get the output value $\hat{y}_1(i,j)$. Then set $\hat{a}(i,j)=0$ and the neighbors as estimated values and perform a convolution with channel model and get $\hat{y}_0(i,j)$. Whichever gives the best match with the actual output data $y(i,j)$ is taken as the update estimate as shown in equation 5-5.

$$\begin{cases} \hat{a}(i,j)=0 & if \quad |\hat{y}_0(i,j)-y(i,j)|<|\hat{y}_1(i,j)-y(i,j)| \\ \hat{a}(i,j)=1 & otherwise \end{cases} \qquad (5\text{-}5)$$

These new estimated values are used for next iteration. The iteration can stop when there is no more change in $\hat{a}(i,j)$ or after a ixed number of iterations. The performance of this detector is as good as that of the LLR for low SNR real data but it is the best for higher SNR simulated data.

5.5 Implementation and Results

The fixed threshold, Adaptive threshold, LLR and iterative detectors are applied to low SNR real data and simulated data of different SNRs. The results for real data and simulated data are shown in Fig. 5-3 and Fig. 5-4, respectively.

The size of the data page is 1024×1280 and because of non-stationarity of the real output pages, the detectors are designed and applied to 64×64 blocks of a page. So each block has its own BER and the page BER is the average of these block-wise BERs over the page. The SNR of real recovered data is around 3db. Simulator is the physical channel model that we have discussed before. To implement the iterative detector we first obtain the 3×3 channel model.

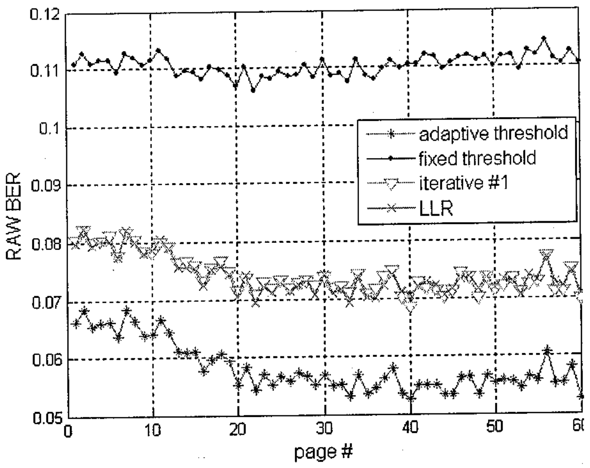


Figure 5-3, BER of Detectors for recovered data

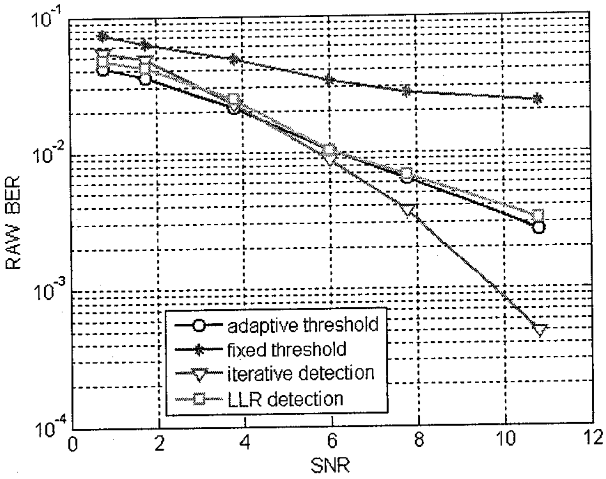


Figure 5-4, BER of Detectors for simulated data

For low SNR real data, the BER performance of the LLR and iterative detector are close. These detectors work better than the fixed threshold but worse than the adaptive threshold. The result of the iterative detector shown is after first iteration. Increasing the number of iteration does not appear to improve the performance of this detector because of error propagation at these high BERs as shown in Fig 5-5. For more iteration, the BER fluctuates around the BER after the first iteration (Fig 5-6).

For higher SNR (SNR>6 db) data, the iterative detector shows the best performance. Fixed threshold detector is the worst detector and has the error floor around 10^{-2}. LLR detector shows a performance close to that of adaptive threshold detector. The result of iterative detector is after 2 iterations. Most of corrections happen during the first and second iteration. Figure 5-7 shows the BER for simulated page at SNR=11 db.

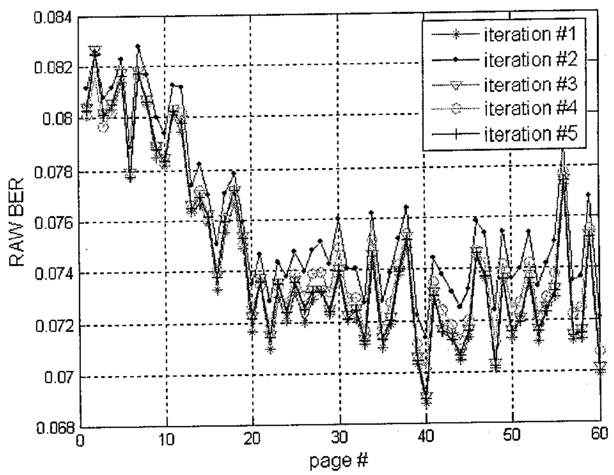


Figure 5-5, First 5 iterations for iterative detector

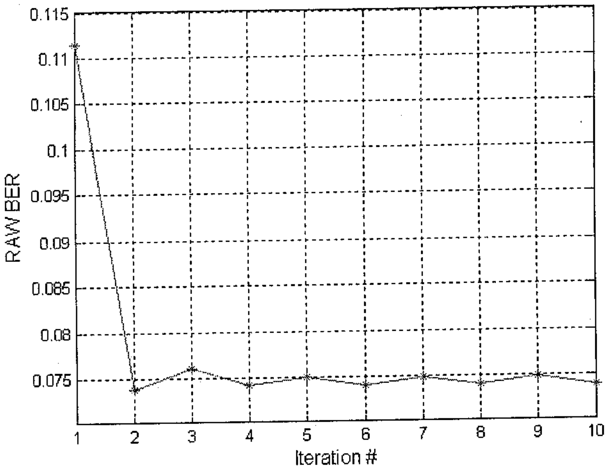


Figure 5-6, Iterative detector for page 30

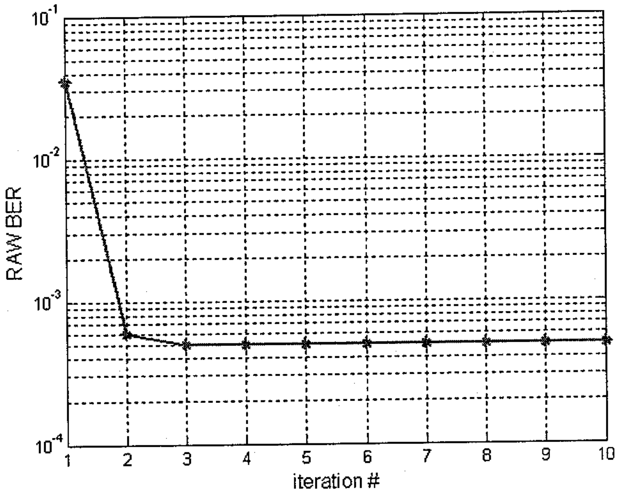


Figure 5-7, Iterative detector for simulated page with SNR= 11 db

5.6 Summary

Fixed threshold detector performs worse than other detectors, but it is a simple detector to implement. For low SNR output page, LLR and iterative detector performances are close but for higher SNR data, the iterative detector works better. Another advantage of iterative detector algorithm is that it is parallel and can be applied to all the output data simultaneously. The disadvantage of iterative detector is that the channel model needs to be known.

Chapter 6

Conclusions and Future Work

6.1 Summary

In this project, our primary objective was to develop a channel model, equalization methods and detection methods for HDS. In this regard we investigated different methods of channel identification, equalization and detection techniques.

For channel modeling we investigated physical channel model, LMMSE channel estimation and look-up-table methods. Physical channel model was developed using physical channel impairment. We compared their performance based on NRMSE using 60 real recorded and retrieved pages, provided by Inphase Technologies. Look-up-table performs better but it is computationally complex and time demanding. The performance of physical channel model is worse than those of others. Its advantage is controlling the impairment amounts that can be utilized to study their impact over the channel.

We investigated the MMSE equalizer, the ZFE, the adaptive decision feedback equalizer and the adaptive Volterra equalizer. MMSE equalizer shows the best performance for real data (with low SNR) and simulated data. The complexity of MMSE equalizer is less than those of other equalizers. Another advantage of MMSE equalizer is that practical and simplified implementations of MMSE equalizer appear to not degrade its performance too much.

For detection methods, fixed threshold, adaptive threshold, log likelihood ratio (LLR) and iterative detectors were investigated. Fixed threshold detector performs worse than other detectors, but it is a simple detector to implement. For low SNR output page, LLR and iterative detector performances are close but for higher SNR data, the iterative detector works better. Iterative detector makes decision with considering neighboring bits and 2D ISI. Another advantage of iterative detector algorithm is that it is parallel and can be applied to all the output data simultaneously. The disadvantage of iterative detector is that the channel model needs to be known.

6.2 Future work

In this project we developed channel model using physical impairments of the channel. The result of this model shows around 30% NRMSE. The reason could be not including all the channel impairments. We also didn't model the storage media. For future we can add more physical channel impairment like misalignment and magnification error. Also we can find a model for storage media too.

In real holographic channel the SNR is very low. As a result the noise enhancement of an equalizer degrades the performance of it. As we can see from the result of real data, the BER improvement is just about 20% and after the equalization the amount of error is still high. Partial response equalizer may be helpful by defining a suitable target and using advanced detection methods.

Because the SNR is very low in HDS, using an equalizer does not improve BER too much. Therefore, to retrieve data reliably, we need to use a powerful coding technique. Different encoding and decoding methods that take into account the 2D nature of channel were not yet investigated and should be considered in the future.

References:

[1] H. Coufal, D. Psaltis, G. Sincerbox (Eds.), *Holographic Data Storage*, (Springer, 2000).

[2] J. Ashley, "Holographic data storage", *IBM J.RES. DEVELOP.*, Vol. 44, pp 341-364 (2000).

[3] L. Agarossi, S. Bellini, P. Migliorati, "Euqalization of Non Linear Optical Channels, " *IEEE International Conference on Communications*, Vol. 2, pp. 662-667 (1998).

[4] L. Ramamoorthy, S. Nabavi, B.V.K. Vijaya Kumar, "Physical Channel Model for Holographic Data Storage Systems," *IEEE Lasers and Electro-Optics Society (LEOS) Conference*, Puerto Rico, Nov 2004.

[5] M. Keskinoz, B.V.K. Vijaya Kumar, "Linear minimum mean squared error (LMMSE) equalization for holographic data storage," *IEEE International Conference on Comunications*, Vol. 3, pp. 1957 – 1961 (1999).

[6] K. M. Chugg, X. Chen, M. A. Neifeld, "Two-Dimensional Linear MMSE Equalization for Page-oriented Optical Memories," *Conference on Signals, Systems & Computers,* Vol. 1, pp. 343-347 (1997).

[7] J. K. Nelson, A. C. Singer, U. Madhow, "Multi-Directional Decision Feedback for 2D Equalization," *IEEE International Conference on Acoustics Speech and Signal Processing*, (ICASSP '04), Vol. 4, pp. 921-924 (2004).

[8] M. Keskinoz and B. V. K. Vijaya Kumar, "Application of linear minimum mean-squared-error equalization for volume holographic data storage," *Applied Optics,* Vol. 38, pp. 4387-4393 (1999).

[9] G. L. Sicuranza, "Quadratic Filters for Signal Processing," *Proceedings of the IEEE*, Vol. 80, pp. 1263-1285 (1992).

[10] E. A. Lee and D.G. Messerschmitt, *Digital Communications*, Kluwer Academic Publishers, Boston, 1988.

[11] John G. Proakis, *Digital communications*, McGraw-Hill, New York, 2001.

[12] M. A. Neifeld, K. M. Chugg, B.M. King, "Parallel data detection in page-oriented memory," *Optics Letters*, Vol. 21, pp. 1481-1483 (1996).

[13] X. Chen, K. M. Chugg, M. A. Neifeld, "Near-optimal Parallel Distributed Data Detection for Page-Oriented Optical Memories," *IEEE Journal of Selected Topics in Quantum Electronics*, Vol. 4, pp. 866-879 (1998).

[14] M. Keskinoz and B. V. K. Vijaya Kumar, "Efficient Modeling and Iterative Magnitude-Squared Decision Feedback Equalization (DFE) for Volume Holographic Storage Channels," *IEEE international Conference on Communications*, Vol. 9, pp. 2696-2700 (2001).